Introduction to JavaScript

- client side scripting language.
- used to make web pages active.
- used to programmatically perform actions within the page.
- When JS was created, it was initial called 'Live Script'.
- But Java was a very popular langu at that time, so it was decide that positioning a language as a 'younger brother' of java would hel

What can JS Do?

- JS can execute not only in the browser, but also on the server
- We will use JS as a client as well as server side language.
- JS has evolved greatly as a language and now used to perform wide variety of tasks.

@ What can't In-Browser JS Do?

- Read/write to from Computer Hard disk.
- ek website doosry website ka data access Nhi kesky.

- What makes JS unique?

- complete integration of Html/CSS.
- Simple things / Simple API.
- Major Modern Browser. (By default Enabled)

- Versions of JS.
- Maintained by community.
- ECMA (standard) → 1997

(47)

JS can be written just before closing tag of body.

```
<script>
    console.log('Hello World');
</script>.
```

(48)

```
<script src="j"> </script>
```

Path of JS file.

```
<script>
  var a = 76;
  var b = "Farzeen";
  console.log(a);
  "        " (b);
```

Dynamic Typing → no need to give datatype

· JS is a Dynamic Typing language

Power
~~Exponent~~ of num1 and 2

console.log("~~Addition~~ is :" + num1 ** num2

concatenation
Power
~~exponent~~ operator

(49) Strings Method in JS.

jb bhi string me single quote (' ') use kna hoga to use double quote ("") se sorround kr denge.
e.g: "Farzee'n";

" " " " double quote "
" " " " single quote " " " "
e.g: 'Farzee"n';

```
var name = "Farzeen";
var message = " is a good boy";
console.log(name+message);
```

jb string me single and double
both quotes use kene hon to
back tick (`) use kerge.
↓
keyboard me esc key k neeche
hain. 1 k barabar

```
var temp = `${name}`;
```
is a nice person

```
var len = name.length;
console.log('length of name is
${len}
);
```
↓
name me jitne letters
honge wo count kerke batadega

```
console.log("Hello\n World");
```
↓
output :- Hello
World

\\ → for show a style \

\n → new line

\b → backspace

\t → tab

(J1)

alert ("Hello world").
consolelog(" " " ");

Agr kisy Id ko target krna
ho to aise krenge:
documentlogetElementById("↓ id name
" " " ".click();
↓
automatically click
hojaega.

<div.class="container">

How to target the class:

container.innerHTML="Hello";

far.innerHTML="Hello";

output → Hello.

document.getElementsByTagName("h1");

- document.write("Faizan");

- console.warn("Ali"); → warning message show krne k liye.

- console.error("error"); → error " " " "

- console.clear(); → console clear krne k liye.

- console.assert(4==6); → koi bhi aisa message show krane k liye jo satisfy na ho to message show hojaeg.

(T2)

Javascript console API:
Application Programming Interface

- console.warn
- " . error
- " . log

Ways to print in JS:

```
console.log("--");
alert("--");
document.write("--");
```

also use this way:

```
console.log("Farzeen", 5+5, "Ali");
```

output: Farzeen 10 Ali

object of JavaScript:

```
console.log({Farzeen: 'this', marks: 34});
```

" .table " " "

output → table ki form me aaega

~~Pehle start kiya time~~

console.time("Your code took");

~~Pehle end kiya time~~

console.timeEnd("Your code took");

~~code start kele kue kar time or end kele kuka time.~~

code ne kitna time lya hi wo btaega.

Variables in JS :-

var f = 23;

there are 3 keyword in JS
- var → old JS me use
- let → modern me use
- const → " " "

const → to making variable const no one ~~could~~ will change its value.
cannot be overwrite.

var has globalscop.

let has blockscope.
↓
can be overwrite.

const → me values add kiskte hain though using array.

```
const arr1 = [12, 10, 9, 7];
arr1.push(99);
console.log(arr1);
```

↓

12, 10, 9, 7 K saath 99 bhi print hoga.

Datatypes:

- Primitive → memory allocat stack.
- Reference → memory allocat heap.

string, numbers, boolean, null, undefined, symbol are types of Primitive D.T.

arrays, object, func., dates are referenc D.T.

- var a name = "Fayen";
- console.log("Datatype is " + (typeof name));

datatype ↓ Staega variable

Datatype of null is object.
↓
Primitive datatype

. Reference :

. myArr = [1, 2, "Ali", false]
↓ string ↓ boolean

console.log("Datatype is" + (typeof myArr));

note: isme bhale koi bhi primitive data type add kro lekin iski datatype object hy rhay y.

(50)

String Functions in JS :

- indexof( ); → use to find the position of a word in string

e.g:

```
var str = "This is String";
```

0 1 2

```
var position = str.indexof ("is");
console.log(position);
```

output = 2 → first occurence

- LastIndexof(); → use to find position of a word from the last occurence of string.

This is string.

0 1 2 3 4 5

```
var position = str.lastIndexof('i');
console.log (position);
```

output = 5 → last occurence

also take -ve value as well as +ve

- slice ( ); → use to print the word in string throug its position.

e.g:

```
var substr = str.slice(1, 6);
console.log(substr);
```

This is a string.

0 ← output → his i

1 5

is
se
ek
kam

aega
bcz 0 se start
nota hi inde

- substring → only take +ve value same as slice.

substring();

substr(); → use to print length of string

for example:

```
var str1 = str.substr(1, 7)
console.log(substr1);
```

starting point of length → 1

end → 7

output :-

his is

1 2 3 4 5 6

- replace( ); → use to replace a word to another word in string.

e.g:

```
var replaced = str.('string', 'fargee');
console.log(replaced);
```

output :- This is fargeen.

str wale variable me changing nhi hogy. jo hamy variable banaya hai replace kaa use shoud hoga just.

- toUpperCase(); → used to convert a string into uppercase letters.

- toLowerCase(); → " " " lowercase "

concat( ); → use for concatination (adding two strings).

e.g:

```
var newString = str.concat('New');
console.log(newString);
```

output - This is a string New.

also use '+' sign for concatination.

- trim( ); → use to remove whitespaces from starting and ending of a string not mid.

- charAt( ); → use to find the char through its index.

e.g:

```
var char2 = str.charAt(2);
console.log (char2);
```

output → (i).

- charCodeAt() → use to print code of char.

for slicing:

console.log(str[2]);

2 index pr jo char hoga wo print hojaega.

(51) Scope and Conditionals:

- var → . can be overwrite
  . scope globally.

let → . cannot be overwrite.
  . scope block.

const → . can't be changed.

if else switch concepts are same as in other languages.

imp

S2 Arrays and Objects:

key value pairs ko store kiya hi.

Object:

```
Let employee={
    name = "Farzeen",
    salary = 10,
channel 1 = "Coding",
        :
        : etc ...
}
console.log(employee);
```

Arrays:

```
Let names = [1, 2, "Ali", undefined];
```

also use:

```
let names = new Array(1, 2, "Ali", undefined);
```

↓ use to making an obj.

```
console.log(names);
console.log(names.length);
" "(names[2]);
```

→ for finding length of an array.

for sorting an Array:

```
names = names.sort();
console.log(names);
```

For adding an element in Array:

```
names.push("fai");
console.log(names);
```

How to acces a key from the Object:

```
employee.channel;
```

Second method:

```
employee["channel 2"];
```

Ye method tb ziyada imp hi jb hamary key k naml me space hoga like ham usko aese use kenge:

e.g:

~~channel 2~~
channel 2" = "Coding",

employee ["Channel 2"]

let names = new Array(23);

This means

23 empty spaces in our Array.

53) Functions in JS:

```
function greet(name)
                 parameter
{
console.log(name + " is a good boy");
}
let name = "f---";
"      "      "
"   "  2"        "
"     3     "     "
"     "     "     "
greet(name);
"      "   1 ;
"     "    2 ;
"      "   3 ;
"     "    " ;
```

for multiple parameters:

```
function (name, greetText)
{
console.log (greetText + " " + name);
    " "  (name + " is a good boy");
}
Let greetText = "Gud Morning";
greet (name, greetText);
```

for default argument:

```
function (name, greetText = "Greetings")
{
    "      "      "
}
```

yeek default argument hi.
by default tb use hoga jb greetText ki value neeche code me initialize nhi ki gae hogy.

Return Value Function:

```
function sum(a,b,c)
{
  let d = a+b+c;
  return d;
}

let returnVal = sum(1,2,3);
console.log(returnVal);
```

output = 6

return k baad koi bhi statement work nhi kaskty.

Interaction using alert();
prompt(); confirm();

Alert:

- used for pop-up message
- takes an argument as string
- Doesnot return anything.

```
alert("This is a Message");
```

Prompt:

- used to take input from the user.

```
let name = prompt("Enter your name",
                  "Guest");
console.log(name);
```

default argument

Confirm:

used for giving warning.

```
let deletePost = confirm("Do u want
                to delete this
                post");
console.log(delete Post);
```

Quiz

```
let age = prompt("Enter your age:"
if (age>18)
{ console.log("u r allowed");
}
else
{ alert("u r not allowed");
}
```

5) Loops in JS

for Loop in JS:

```
Let friends = ["farzeen", "Aman", "AF"];

for (let index=0; index<friends.length;
        index++)
  {
    console.log("Hello," + friends[index]);
```

Using Modern JS:

for Each Loop:

```
friends.forEach(function f(element)
  {
console.log("Hello," + element+" to modern JS");
  });
```

for of Loop:

```
for (element of friends)
{ console.log("Hello,"+element+" to modern JS");
  }
```

- Use this loop to iterate over objects in JavaScript:

```
let employee = {
name = "Fauzeen",
salary = 200000,
channel = "CwF",
}

for (key in employee)
{
console.log(`The ${key} of employee is
${employee[key]}`);
}
```

- While loop in JS:

```
let i = 0;
while (i<4)
{
console.log(`${i} is less than 4`);
i++;
}
```

Do-while loop:

```
let i=0;
do
{
console.log(`${i} is less than 4`);
    i++;
}
while(i<4);
```

6) Manipulating DOM:

DOM → Document Object Model
↓
Html code.
write in browser:
nav.InnerHTML press enter
you will get nav HTML code
to change inner HTML:
nav.InnerHTML = "<li>Dynamic</li>".
to grab element by class name

```
et containers = document.getElementsByClassName('container');
console.log(containers);
```

To grab CSS selector:
e.g: .container, #nav, ...

```
let sel = document.querySelector('.container');
console.log("Selector returns ", sel);
```

↓

all types of selector are grab like this.

to grab All matching selector element:

```
Let sels = document.querySelectorAll('#nav>li');
console.log("Selector returns", sels);
```

(57) Events:

Event mean jb bhi ham btn pr click kren ya koi bhi action perform kren to uska reaction kya hoga.

e.g:
we create a h2, p with id = "para" and button with id = "btn".

```
<button id = "btn" onclick = "togglehide()">
                Show/hide </button>

<script>
function togglehide()
 {
let btn = document.getElementbyId('btn'
" para = " " " ('para');

if(para.style.display != 'none') {

para.style.display = 'none';
}
 else {
  " " " = 'block';
 }
}
```

MouseOver / MouseOut

```
let para = document.getElementbyId('para');
para.addEventListener('mouseover', function run() {
  consolelog ("MouseInside");
  });
" " " " ('mouseout', " "
" " ("MouseOutside");
```

JS and [illegible] setInterval k bajae ye use kiya jaata hai:
window.requestAnimationFrame(main);
a method passed as an argument.

(58) Set timeout Clear Timeout:

Set timeout → Allows us to run the function once after the interval of time.

Clear Timeout → Allows us to run the function repeatedly after the interval of time.

e.g of Set Timeout:

```
function greet()
  { console.log("Good Morning"); }

setTimeout(greet, 5000);
```

name likha is just func call ki kaya trick

func. name

time interval in millisecs.

for multiple arguments:

```
function greet(name, byeText)
{console.log("Hello"+name+" "+byeText);
setTimeOut(greet, 5000, "Faizeen", "bye");
```

```
timeOut = SetTimeout(greet, 5000, "Ali", "bye");
console.log(timeOut);
```

↓

Ye ek unique id return kregA.

```
ClearTimeOut(timeOut);
```

↓

iski best eg ye hi k agr ham chalte hain k koi hothay website pr ho or ek time interval k baad ham use sign in ka alert show kraen or agr user us se pehle hy sign in kr de to us time ko clear krne k lye ye use hoge.

SetInterval:

```
et Interval(greet, 1000, "Ali", " Bye");
```

↓

same output print kita rhega apne time interval k baad baar baar.

```
nterval Id = " " " " ";
clearTimeout = (intervalId);
```

Making Clock:

Time now is

```
<script>
function displayTime() {
time = new Date();
console.log(time);
document.getElementById('time').innerHTML = ti
}
setInterval(displayTime, 1000);
```

59 Date and Time

Let now = new Date(); →
Date is a constructor and its type is Object.

```
let dt = new Date(20);
consde.log(dt);
```
↳ ek reference time de deta hi

How to intialize data:

```
let newDate = new Date("2021-10-6");
console.log(newDate);
```

Another way:

```
// let newDate = new Date (year, month, date, hour, min, sec
                                                , ms);

let newDate = new Date(2021, 8, 6, 7, 21, 2, 35);
console.log(newDate);
```

// newDate K variables me jo year month date etc hain unko separately access aese kiskte hain:

```
it yr = newDate.getFullYear();
console.log("The year is:" yr);

" date = " .getDate();
"      "  " date "  date);

let month = " .getMonth();
"     "   month , month);
```

```
newDate.setDate(7);
```
→ date to set kiye even wo invalid q? na ho

```
"  " .setSeconds(22);
```

6

For Making Clock:

Html:

```
<div class = container>
Current time is <span id = "time"> </span>
</div>
```

JS:

```
setInterval(updateTime, 1000);
```

1s

```
function updateTime()
{
time.innerHTML = new Date();
}
```

For rotation css with JS:

```
hour.style.transform = 'rotate(${variable of rotation}deg)';
```

Id — CSS — Property — value — var for rotation

60) Arrow functions in JS:

It is a shortcut to define ~~Arrow~~ functions.

Syntax:

```
let greet = () => {
    console.log("Hello");
  };
greet();
```

Another shortcut:

```
let greet = () => console.log("Hi");
```

// This is used when no argument. agr argument na ho tb bhi bracket khali lgani pregy.

```
let sum = (a,b) => {
    return (a+b);
  };
```

*

important thing:

```
Let obj1 = {
greeting: "Good Morning",
names: ["Faiqeen", "Amlan", "AF"],

speak(){
  this.names.forEach((student)=>{
Consolelog("Hi"+ this.greeting+" "+student)
  });

}
}
obj1.speak();
```

this → allow func. me this ka matlab uska parent like jis k andar wo hi from the above e.g speak is inside the obj1 object.

- Agr hum speak k func. me simple func. bnate allow func. thi bnate to output undefine aata bcz

this method also known as "lexical this."

Simple func me agr ham 'this' use karenge to wo apne andar hy 'this' find karega 'parent' me nhi jaega or allow func. ~~ath~~ kmesha parent ka 'this' find krta hai.

---

let sum2 = (a, b) => a+b;
arg. → return value

// write only when we have two arguments and 1 return value.

let half = a => a/2;
// write only when we have 1 argument

above functions ko runtime pr aese use karenge'
Sum2 (value of a, value of b)
Press enter
You will get the answer.

(61) Math Object:

- For printing the Math Object:

```
let m = Math;
console.log(m);
```

jitne bhi math ~~K~~ objects hote hain like power, sqrt, sin, cos tan etc sb aajaenge.

- For printing the constants:

```
console.log("The value of Math.PI is", Math.PI);
" " " " Math.LN10 " , Math.LN10);
```

- For printing the functions from mathobject

```
let a = 34.65;
let b = 49;
console.log(Math.round(a), Math.round(b));
```

for round of

Math.pow(3, 2) → 3 ki power 2

" . sqrt(36) → sqrt of 36

Math.ceil(5.8) → returns rounded upto nearest integer

Math.floor(5.8); → returns rounded down to nearest integer.

Math.abs(-5.6); → always returns in +ve value.

Math.sin(Math.PI);

" cos " "

" tan " "

Important

Ye random no 0 to 1 k b/w generate kega:

let r = Math.random();

console.log("The Random no is:", r);

Agr 1 se 100 k b/w chahiye to ye formula use hoga:

let a1 = 1;

let b1 = 100;

∵ Random no b/w (a, b) = a + (b-a) * Math.Random();

let r1_100 = a1 + (b1-a1) * Math.Random();

console.log("Random no b/w 1 to 100 is:", r1_100);

62

JSON → JavaScript Object Notation.

- used for transport your data.

```
Let jsonObj = {
 name : "Farjeen",
 channel : "C>F",
 friend : "Afrid",
 food : "Biryani",
 }
console.log( jsonObj );
```

- below syntax for obj to convert into string :

```
let myJsonStr = JSON.stringify(jsonObj);
console.log (myJsonStr) ;
```

- below syntax for replace :

```
myJsonStr = myJsonStr. replace("Farjeen", "Afrid")
console.log (myJsonStr);
```

- below syntax is used for valid string convert into obj

```
newJsonObj = JSON.parse(myJsonStr);
Console. Log( newJsonObj );
```

JSON Tutorial:

- donot use single quote '' in JSON.
u can use nos, strings, null, boolean, arrays, objs, nested objs etc.

create file
index.json.

```
{
 "name" : "Farzeen",
 "score" : 25.5,
 "isAdmin" : false,
 "Lisence" : null,
 "shop2" : ["food", "clothes", {"a": true}],
 "myObj" : {
      "name" : "Aman",
      "marks" : 34
    }
}
```

- for access in console:
- myJSON = '{ copy paste }'

press enter.

- JSON.parse(myJSON);
- parsed["shop2"][1] → press enter you will get "clothes".